THEORETICAL CHEMISTRY INSTITUTE

THE UNIVERSITY OF WISCON

UNPUBLISHED PRELIMINARY DATA

A NEW ALGORITHM FOR QUADRATURE

by

Gunter G. Weber

WIS-TCI-79 26 March 1965

N65 23601
(ACCESSION NUMBER)
20
(PAGES)
CR-62662
(NASA CR OR TMX OR AD NUMBER)
1
(THRU)
(CODE)
19
(CATEGORY)
FACILITY FORM 602

GPO PRICE $
OTS PRICE(S) $
Hard copy (HC) 1.00
Microfiche (MF) .50

MADISON, WISCONSIN

A NEW ALGORITHM FOR QUADRATURE *

by

Gunter G. Weber

Theoretical Chemistry Institute, University of Wisconsin
Madison, Wisconsin 53706

ABSTRACT

23661

A new algorithm for interpolation, inverse interpolation, and integration of functions given by a table is proposed and numerical examples are given.

author

* Work supported by the National Aeronautics and Space Administration, Grant NsG-275-62.

Introduction

The approximate integration of functions of one variable falls into two classes: definite integrals and indefinite integrals.** It can be easily seen that this classification is rather arbitrary. Nevertheless, we could say that e.g. the Gauss integration

$$\int_{-1}^{+1} f(x)dx \approx \sum_k A_k^{(n)} f(x_k^{(n)})$$

belongs to the first class if f(x) is given analytically, but the integration of functions given in tabular form belongs to the second class. We describe a new algorithm for quadrature which works also if the values of the table are nonequidistant.

It is correlated to the Aitken-Interpolation[1] and can also be used (with slight modifications) for interpolation and inverse interpolation.

The Conventional Aitken-Interpolation

Let f(x) be a function of x, and let $f_j = f(x_j)$, $f_k = f(x_k)$. Then we can write the linear interpolation

$$f_{jk}(x) = \frac{\begin{vmatrix} f_j & x_j - x \\ f_k & x_k - x \end{vmatrix}}{x_k - x_j}$$

- - - - -

** A division in which we follow V. I. Krylov's "Approximate Calculation of Integrals", ACM-Monograph Series, MacMillan Company, New York, translated by A. H. Stroud, 1962 (Moscow 1959).

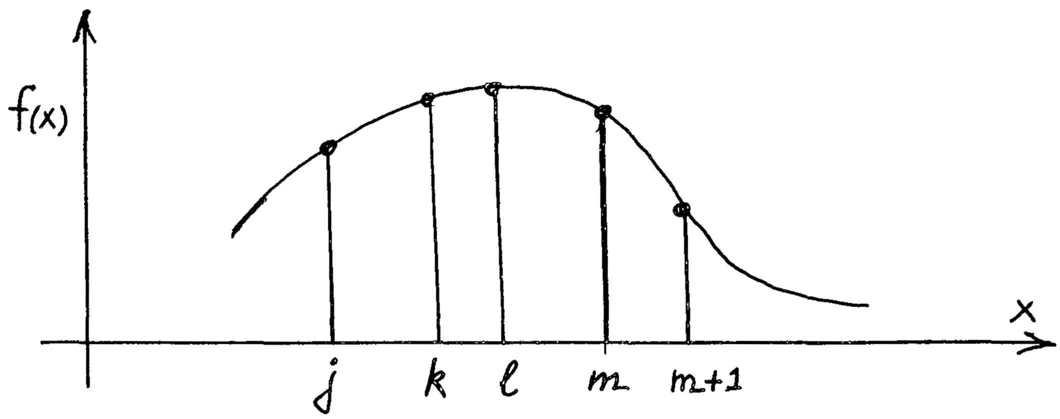
In an analogous way we can calculate the interpolation of n-th degree by n-times iterating a linear interpolation

$$f_{j_1 j_2 \cdots j_{n+1}}(x) = \frac{\begin{vmatrix} f_{j_1 j_2 \cdots j_n} & x_{j_1} - x \\ f_{j_2 j_3 \cdots j_{n+1}} & x_{j_{n+1}} - x \end{vmatrix}}{x_{j_{n+1}} - x_{j_1}}$$

For more information see reference 2.

The Moving Aitken-Interpolation

Let f(x) be a function to be interpolated. It is well known, that high degree polynomials - even if we could disregard the difficulty to evaluate them - still might give a strongly oscillating behavior.



It is therefore reasonable to try the method for a low degree polynomial. For example, we chose a quadruple of points, (a generalization to n is given in Appendix I)

$$\{ j, k, \ell, m \}$$

which define, preferably (but not necessarily) between k and ℓ the 3rd order polynomial. Then we go over to the next set of four points

$$\left\{k, \ell, m, m+1\right\}$$

and rename them

$$\left\{j, k, \ell, m\right\}$$

Now we describe in more detailed fashion, what we are doing using a symbolic language[3] and a Legend which should be easily understood. Notice: in the Appendix IV we also give a FORTRAN-63-PROGRAM for a similar process, the quadrature.

Moving Aitken-Loop

Legend

$\{j, k, \ell, m\} \xleftarrow{\text{loop}}$ — choose 4 consecutive points

$\{f_j, f_k, f_\ell, f_m\}$ — calculate the function values

$f_{jk} \longleftarrow \{f_j, f_k; x_j, x_k\}$

$f_{k\ell} \longleftarrow \{f_k, f_\ell; x_k, x_\ell\}$

$f_{\ell m} \longleftarrow \{f_\ell, f_m; x_\ell, x_m\}$

linear interpolations define the f_{jk}, $f_{k\ell}$, and $f_{\ell m}$

$\to f_{jk}$ (stored)

$\to f_{k\ell}$ (stored)

$\to f_{\ell m}$ (stored)

$f_{jk\ell} \longleftarrow \{f_{jk}, f_{k\ell}, x_j, x_\ell\}$

$f_{k\ell m} \longleftarrow \{f_{k\ell}, f_{\ell m}; x_k, x_m\}$

linear interpolations define $f_{jk\ell}$ and $f_{k\ell m}$

$f_{jk\ell m} \longleftarrow \{f_{jk\ell}, f_{k\ell m}; x_j, x_m\}$

linear interpolation defines $f_{jk\ell m}$

$f_{jk\ell m}(x)$ (stored)

PRINT $\longleftarrow f_{jk\ell m}(x_\mu) \longleftarrow \boxed{\text{READ IN } x_\mu}$

$j + 1 \longleftarrow j$

$k + 1 \longleftarrow k$

$\ell + 1 \longleftarrow \ell$

$m + 1 \longleftarrow m$

move the indices

$m : N \xrightarrow{m < N \text{ loop}}$

$\downarrow$ $m \geqslant N$

STOP

comparison of m with the largest index N available

We will give in Appendix II examples where the Moving AITKEN has been applied.

Interpolation of a Tabulated Function and Inverse Interpolation

We bring these subjects together since for an inverse interpolation a tabulated function would be an unusual simplification. For, if we have a function u(v) tabulated

$$\left\{ v_\mu \quad , u_\mu \right\}$$, (where the bracket as usual designates a set),

the inverse function v(u) is just

$$\left\{ u_\mu \quad , v_\mu \right\} ,$$

in other words, a permutation of the rows. This technique would change nothing of the structure of the Aitken-Loop only that the function-values now work as variables and that the variables now work as function values. An example of this technique is shown in Appendix III where the root is found by reading in the new variable zero, giving "y_{root}" as function value of the inverse function.

Moving AITKEN in Different Representations

The Moving Aitken can give us a set of polynomials, each valid for a small part of the function to be integrated. This means that we can integrate without the usual correction term. The only error arises from the Aitken-Interpolation. Even this situation can give us very different precisions, depending on the precision possible for the interpolation. Some examples are given in Appendix V.

Let us also state that we can express our 3^{rd} degree polynomial by a matrix $\underset{\sim}{f}_{jk\ell m}$ with the property

$$\det \underset{\sim}{f}_{jk\ell m} = f_{jk\ell m}$$

where $f_{jk\ell m}$ is another way of writing the 4-point interpolation. $\underset{\sim}{f}_{jk\ell m}$ is shown on page 7 and $f_{jk\ell m}$ on page 8. We will then find a formula easier to integrate on the consecutive pages.

MOVING-[illegible]EN-MATRIX

$$
\underset{\sim}{f}_{jk\ell m} = \begin{pmatrix}
\frac{f_j}{(x_m-x_j)(x_\ell-x_j)(x_k-x_j)} & x_j-x & x_j-x & 0 & x_j-x & 0 & 0 & 0 \\
\frac{f_k}{(x_m-x_j)(x_\ell-x_j)(x_k-x_j)} & x_k-x & 0 & 1 & 0 & 1 & 0 & 0 \\
\frac{f_k}{(x_m-x_j)(x_\ell-x_j)(x_\ell-x_k)} & x_k-x & x_\ell-x & 0 & 0 & 0 & 1 & 0 \\
\frac{f_\ell}{(x_m-x_j)(x_\ell-x_j)(x_\ell-x_k)} & x_\ell-x & 0 & 1 & 0 & 0 & 0 & 1 \\
\frac{f_k}{(x_m-x_j)(x_m-x_k)(x_\ell-x_k)} & x_k-x & x_k-x & 0 & x_m-x & 0 & 0 & 0 \\
\frac{f_\ell}{(x_m-x_j)(x_m-x_k)(x_\ell-x_k)} & x_\ell-x & 0 & 1 & 0 & 1 & 0 & 0 \\
\frac{f_\ell}{(x_m-x_j)(x_m-x_k)(x_m-x_\ell)} & x_\ell-x & x_m-x & 0 & 0 & 0 & 1 & 0 \\
\frac{f_m}{(x_m-x_j)(x_m-x_k)(x_m-x_\ell)} & x_m-x & 0 & 1 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

MOVING-AITKEN-DETERMINANT

$$f_{jk\ell m} = \begin{vmatrix} \begin{vmatrix} \begin{vmatrix} \dfrac{f_j}{(x_m-x_j)(x_\ell-x_j)(x_k-x_j)}, & x_j-x \\ \dfrac{f_k}{(x_m-x_j)(x_\ell-x_j)(x_k-x_j)}, & x_k-x \end{vmatrix} & , \quad x_j-x \\ \begin{vmatrix} \dfrac{f_k}{(x_m-x_j)(x_\ell-x_j)(x_\ell-x_k)}, & x_k-x \\ \dfrac{f_\ell}{(x_m-x_j)(x_\ell-x_j)(x_\ell-x_k)}, & x_\ell-x \end{vmatrix} & , \quad x_\ell-x \end{vmatrix} & , \quad x_j-x \\ \begin{vmatrix} \begin{vmatrix} \dfrac{f_k}{(x_m-x_j)(x_m-x_k)(x_\ell-x_k)}, & x_k-x \\ \dfrac{f_\ell}{(x_m-x_j)(x_m-x_k)(x_\ell-x_k)}, & x_\ell-x \end{vmatrix} & , \quad x_k-x \\ \begin{vmatrix} \dfrac{f_\ell}{(x_m-x_j)(x_m-x_k)(x_m-x_\ell)}, & x_\ell-x \\ \dfrac{f_m}{(x_m-x_j)(x_m-x_k)(x_m-x_\ell)}, & x_m-x \end{vmatrix} & , \quad x_m-x \end{vmatrix} & , \quad x_m-x \end{vmatrix}$$

The Integral by a Purely Arithmetic Procedure

Now we will derive the computational form of the integral

For a program see Appendix IV. Two simple rules of linear algebra are needed.

a) Additivity of Columns

$$\begin{vmatrix} a_{11} + b_{11} & a_{12} \\ a_{21} + b_{21} & a_{22} \end{vmatrix} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} + \begin{vmatrix} b_{11} & a_{12} \\ b_{21} & a_{22} \end{vmatrix}$$

b) Multiplication of Columns

$$\begin{vmatrix} a_{11} & c \\ a_{21} & c \end{vmatrix} = \begin{vmatrix} a_{11} & 1 \\ a_{21} & 1 \end{vmatrix} c$$

This will give for the linear interpolation

$$f_{jk}(x) = \frac{\begin{vmatrix} f_j & x_j \\ f_k & x_k \end{vmatrix} - \begin{vmatrix} f_j & 1 \\ f_k & 1 \end{vmatrix} x}{x_k - x\,;} = f^{(0)}_{jk} + f^{(1)}_{jk}\, x$$

Similarly we get $f_{jk\ell m}$ and $\int_{x_k}^{x_\ell} f_{jk\ell m}\, dx$. The four-point interpolation can be written

$$f_{jk\ell m}(x) = \frac{\begin{vmatrix} f_{jk\ell} & x_j - x \\ f_{k\ell m} & x_m - x \end{vmatrix}}{x_m - x_j}$$

$$= \frac{\begin{vmatrix} f^{(0)}_{jk\ell} + f^{(1)}_{jk\ell}\, x + f^{(2)}_{jk\ell}\, x^2 & x_j - x \\ f^{(0)}_{k\ell m} \quad f^{(1)}_{k\ell m}\, x + f^{(2)}_{k\ell m}\, x^2 & x_m - x \end{vmatrix}}{x_m - x_j}$$

Where the $f^{(0)}_{jk\ell}$, $f^{(1)}_{jk\ell}$, $f^{(2)}_{jk\ell}$ terms are defined as

$$f_{jk\ell}(x) = f^{(0)}_{jk\ell} + f^{(1)}_{jk\ell}\, x + f^{(2)}_{jk\ell}\, x^2$$

$$= \frac{\begin{vmatrix} f^{(0)}_{jk} & x_j \\ f^{(0)}_{k\ell} & x_\ell \end{vmatrix} + \left\{ \begin{vmatrix} f^{(1)}_{jk} & x_j \\ f^{(1)}_{k\ell} & x \end{vmatrix} - \begin{vmatrix} f^{(0)}_{jk} & 1 \\ f^{(0)}_{k\ell} & 1 \end{vmatrix} \right\} x - \begin{vmatrix} f^{(1)}_{jk} & 1 \\ f^{(1)}_{k\ell} & 1 \end{vmatrix} x^2}{x_\ell - x_j}$$

and $f_{k\ell m}(x)$ in an analogous way (with $f^{(0)}_{k\ell m}$, $f^{(1)}_{k\ell m}$, $f^{(2)}_{k\ell m}$). The next lower step is $f_{jk}(x)$ and $f_{k\ell}$, $f_{\ell m}$). It can be written

$$f_{jk} = f^{(0)}_{jk} + f^{(1)}_{jk}\, x$$

This concludes our derivation of $f_{jk\ell m}$. Again applying the mentioned rules of linear algebra, we can now write $f_{jk\ell m}$ explicitly.

$$f_{jk\ell m}(x) = \frac{\begin{vmatrix} f^{(0)}_{jk\ell} & x_j \\ f_{k\ell m} & x_m \end{vmatrix} + \left\{ \begin{vmatrix} f^{(1)}_{jk\ell} & x_j \\ f^{(1)}_{k\ell m} & x_m \end{vmatrix} - \begin{vmatrix} f^{(0)}_{jk\ell} & 1 \\ f^{(0)}_{k\ell m} & 1 \end{vmatrix} \right\} x + \left\{ \begin{vmatrix} f^{(2)}_{jk\ell} & x_j \\ f^{(2)}_{k\ell m} & x_m \end{vmatrix} - \begin{vmatrix} f^{(1)}_{jk\ell} & 1 \\ f^{(1)}_{k\ell m} & 1 \end{vmatrix} \right\} x^2 - \begin{vmatrix} f^{(2)}_{jk\ell} & 1 \\ f^{(2)}_{k\ell m} & 1 \end{vmatrix} x^3}{x_m - x_j}$$

$$= f^{(0)}_{jk\ell m} + f^{(1)}_{jk\ell m}\, x + f^{(2)}_{jk\ell m}\, x^2 + f^{(3)}_{jk\ell m}\, x^3$$

These coefficients should be easy to calculate by application of the foregoing formulas. They also should reduce the integration to a simple procedure:

$$\int_{x_k}^{x_\ell} f_{jk\ell m}(x)\,dx = f^{(0)}_{jk\ell m}(x_\ell - x_k) + f^{(1)}_{jk\ell m}\frac{(x_\ell^2 - x_k^2)}{2} + f^{(2)}_{jk\ell m}\frac{(x_\ell^3 - x_k^3)}{3} + f^{(3)}_{jk\ell m}\frac{(x_\ell^4 - x_k^4)}{4}$$

Some examples will be given in Appendix V. It goes without saying that there are situations where a lower degree formula has to be accepted.

Acknowledgment

The coding of the computational program was carried out by Mrs. Merline McCloud and Mrs. Carol Monash.

Appendix I

The method outlined above can be generalized as follows:

$$f_{j_1 \cdots j_n} =$$

$$\frac{1}{x_{j_n} - x_{j_1}} \left\{ \begin{vmatrix} f^{(0)}_{j_1 \cdots j_{n-1}} & x_{j_1} \\ f^{(0)}_{j_2 \cdots j_n} & x_{j_n} \end{vmatrix} - \begin{vmatrix} f^{(n-2)}_{j_1 \cdots j_{n-1}} & 1 \\ f^{(n-2)}_{j_2 \cdots j_n} & 1 \end{vmatrix} X^{n-1} \right\}$$

$$+ \frac{1}{x_{j_n} - x_{j_1}} \sum_{\nu=1}^{n-2} \left\{ \begin{vmatrix} f^{(\nu)}_{j_1 \cdots j_{n-1}} & x_{j_1} \\ f^{(\nu)}_{j_2 \cdots j_n} & x_{j_n} \end{vmatrix} - \begin{vmatrix} f^{(\nu-1)}_{j_1 \cdots j_{n-1}} & 1 \\ f^{(\nu-1)}_{j \cdots j_{n-1}} & 1 \end{vmatrix} \right\} X^{\nu}$$

Appendix II

Examples of Moving Aitken-Interpolation

We took four points, e.g.

2.0

2.1

2.2

2.3

and interpolated cosh x with x = 2.15, then went on to another quadruple of points, e.g.

2.1

2.2

2.3

2.4

and interpolated cosh x with x = 2.25; some interpolated values are listed (the line indicates the decimals we got.)

x	cosh(x μ)
2.1500000	4.35066107208
2.2500000	4.79655627913
2.3500000	5.29045703359
2.4500000	5.83730646023
2.5500000	6.44257761191
2.6500000	7.11232824575
2.7500000	7.85326145122

Note that the "Moving Aitken" is without any change applicable to nonequidistant points. (See also Appendix III).

Appendix III.

Example for "Inverse Interpolation"

The inverse interpolation is done by inverting the two columns giving the variable x, and the function value f(x). The f(x) values are taken as (nonequidistant) variables* and the x values are taken as function values.*

Variable f(x)	Function-Value x			
-.0548663432	.2857142857			
.6144008937	.2777777777	.2850636529		
1.0388062561	.2702702702	.2886462062	.2852433791	
1.0982165601	.2941176470	.2818624741	.2862125448	.2851924139

$.28519241394 = x_{root}$, i.e. $f(x_{root}) = 0$

- - - - -

* Of the inverse function of f(x).

Appendix IV.

Subroutine Aitken

```
      SUBROUTINE AITKEN(LIML,LIMU)
      DIMENSION COEF(10,10),SAVE(10,10)
      COMMON/USE2/AREAP, GINT(10), YINT(10), INTD, COEF
      DO 1 M=1,INTD
 1    COEF(M,1)=GINT(M)
      NCOEF=1
      LIM=INTD
 8    LIM=LIM-1
      IF(LIM)2,2,3
 3    NCOEF=NCOEF+1
      M=1
 7    INDEX=NCOEF+M-1
      DENOM=YINT(INDEX)-YINT(M)
      SAVE(M,1)=(COEF(M,1)*YINT(INDEX)-COEF(M+1,1)*YINT(M))/DENOM
      N=2
 6    IF(N-NCOEF+1)4,4,5
 4    SAVE(M,N)=(COEF(M,N)*YINT(INDEX)- COEF(M+1,N)*YINT(M)-COEF(M,N-1)
     1 +COEF(M+1,N-1))/DENOM
      N=N+1
      GO TO 6
 5    SAVE(M,N)=(-COEF(M,N-1)+COEF(M+1,N-1))/DENOM
      M=M+1
      IF(M-LIM)7,7,13
 13   DO 14 M=1,LIM
      DO 14 N=1,NCOEF
 14   COEF(M,N)=SAVE(M,N)
      GO TO 8
      ENTRY AITKENP
 2    YU=1.
      YL=1.
      DIV=0.
      AREAP=0.
      DO 9 N=1, NCOEF
      YU=YU*YINT(LIMU)
      YL=YL*YINT(LIML)
      DIV=DIV+1.
 9    AREAP=AREAP+COEF(1,N)*(YU-YL)/DIV
      PRINT 12, AREAP
 12   FORMAT(7H AREAP=E17.10)
      RETURN
      END
```

The "Subroutine Aitken" has been used in many cases, a few of them are reported here. It integrates a tabulated function between two points which can be read in (AITKENP). The degree of the polynomial representing the integrand is variable. The definite integral thus calculated is called AREAP.

Appendix V.

Numerical Results

(1) An integration of

$$-\int_0^{2\pi} \cos x dx$$

was done. The integral is of the order of 10^{-7}, in other words it is correct to 6 decimals. The stepwidth was generally .1

(2) An integration of a function f(x) was done

(a) by its analytical formula and by 32-point-Gauss,

(b) by four points (j, k, ℓ, m) lying on the curve - unequally spaced - and by the proposed algorithm.

Gauss evidently must be taken as "standard" of the comparison.

Integral by Gauss	Integral by Aitken
$1.3606219962 \times 10^{-2}$	$1.3606258427 \times 10^{-2}$
$1.7660011005 \times 10^{-2}$	$1.7660105432 \times 10^{-2}$
$2.3272151107 \times 10^{-2}$	$2.3272396486 \times 10^{-2}$

The integration region is about .01. We have 5 decimals agreement. Of course the correlation can get much worse if the integration limits are too far apart or if the curve gets too steep; however, each method deteriorates here.

References

1. A. C. Aitken, Proc. Edin. Math. Soc. Ser 2, 3, (1932), 56, and J. Todd, Survey of Numerical Analysis, McGraw Hill Book Company, Inc., 1962.

2. Reference 1, and N. Macon, Numerical Analysis, John Wiley and Sons, Inc., New York, 1963.

3. We think that for the description of such an algorithm (not the actual program) K. E. Iverson, A Programming Language, John Wiley and Sons, Inc., New York, 1962, might be a good example.